

- Filtros não lineares:

- Estatísticos

- Mediana
 - Moda

- Derivadas e gradientes

- Robert
 - Sobel
 - Prewitt

->Declividade e Aspecto

- Morfológicos

- Erosão e Dilatação

Filtros estatísticos

- Mediana

- Moda

$X = [0 \ 1 \ 2 \ 2 \ 3 \ 4 \ 5 \ 6 \ 9]$

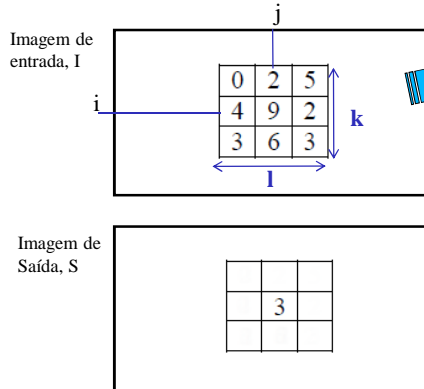
`median()`

`mode()` % não tem no Freemat

Filtro da mediana

- A aplicação da máscara de mediana consiste em:
 - Ordenar os valores de $I(i,j)$ e vizinhos contidos na máscara centrada em (i,j) .
 - Substituir o valor $I(i,j)$ pelo valor da mediana.

Implementação do Filtro da mediana 3 x 3:



```
df = 3 % tamanho do filtro
d1 = floor(df/2);
d2 = ceil(df/2);
```

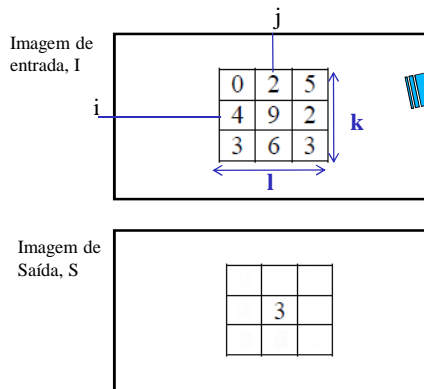
```
vetor(1) = 0;
vetor(2) = 2;
vetor(3) = 5;
vetor(4) = 4;
vetor(5) = 9;
vetor(6) = 2;
vetor(7) = 3;
vetor(8) = 6;
vetor(9) = 3;
```

% para cada posição (i,j) da imagem I, criar vetor

Filtro da mediana

- A aplicação da máscara de mediana consiste em:
 - Ordenar os valores de $I(i,j)$ e vizinhos contidos na máscara centrada em (i,j) .
 - Substituir o valor $I(i,j)$ pelo valor da mediana.

Implementação do Filtro da mediana 3 x 3:



```
df = 3 % tamanho do filtro
d1 = floor(df/2);
d2 = ceil(df/2);
```

```
vetor(1) = 0;
vetor(2) = 2;
vetor(3) = 5;
vetor(4) = 4;
vetor(5) = 9;
vetor(6) = 2;
vetor(7) = 3;
vetor(8) = 6;
vetor(9) = 3;
```

```
% para cada posição (i,j) da imagem I, criar vetor:
for k = -d1:d1
  for l = -d1:d1
    vetor(df*(d2+k-1)+(d2+l)) = I(i+k, j+l);
  end
end
S(i,j) = median(vetor);
```

Filtro da moda

- Seleciona o valor que ocorre com maior frequência na vizinhança para substituir o valor do pixel central.

1	1	2
1	4	1
1	1	1



1

```
nclasses = 5; %numero de classes
mf = 3 % tamanho do filtro
d1 = floor (mf/2);
d2 = ceil(mf/2);
```

% para cada posição do filtro efetuar contagem em vetor:

```
vetor(1) = 6;
vetor(2) = 2;
vetor(3) = 0;
vetor(4) = 1;
vetor(5) = 0;
```

```
% encontrar a posição maior valor
posicao = find(vetor == max(vetor));
S(i,j) = posicao;
```

• Exemplo de aplicação:

O filtro da moda permite melhorar os resultados da segmentação e da classificação de imagens, tornando as regiões mais homogêneas.

Filtro da moda

- Seleciona o valor que ocorre com maior frequência na vizinhança para substituir o valor do pixel central.

1	1	2
1	4	1
1	1	1



1

```
nclasses = 5; %numero de classes
mf = 3 % tamanho do filtro
d1 = floor (mf/2);
d2 = ceil(mf/2);
```

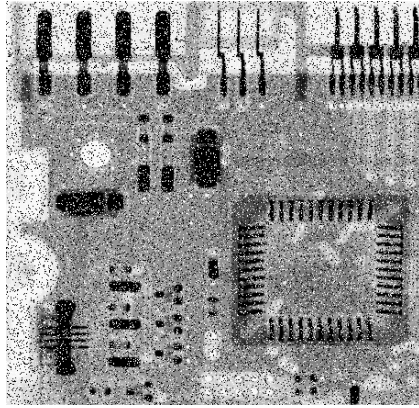
% para cada posição do filtro efetuar contagem em vetor:

```
vetor = zeros(nclasses,1);
for k = -d1:d1
for l = -d1:d1
    valor = I(i+k, j+l);
    vetor(valor,1) = vetor(valor,1)+1;
end
end
% encontrar a posição maior valor
posicao = find(vetor == max(vetor));
S(i,j) = posicao;
```

• Exemplo de aplicação:

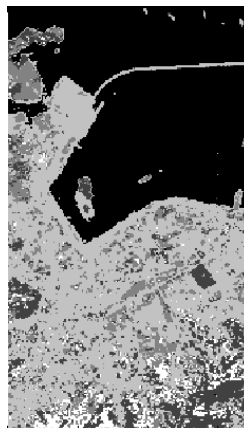
O filtro da moda permite melhorar os resultados da segmentação e da classificação de imagens, tornando as regiões mais homogêneas.

Exercício 1:
Minimizar o ruído da imagem 'Sal_e_Pimenta.tif'.



Exercício 2: A imagem temática resultante da classificação apresenta pequenas fragmentos no interior das classes. Melhore o resultado da classificação.

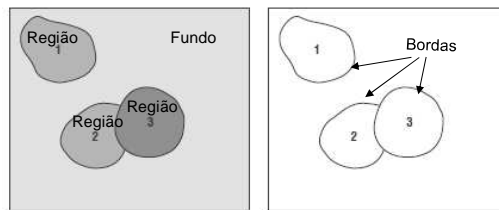
Imagem
resultante da
classificação:
Rio_classes.tif



Classes:
1- água
2 – vegetação
3, 4 – urbano
5 – solo
exposto

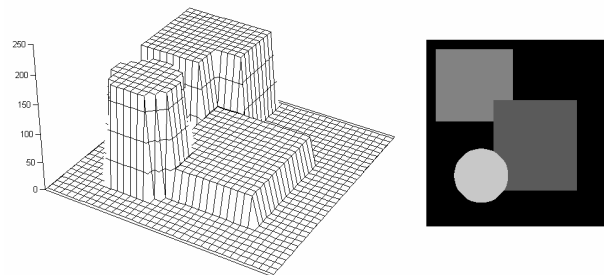
Derivada e gradiente

-> Implementação de derivadas em filtros que ressaltam bordas e transições .

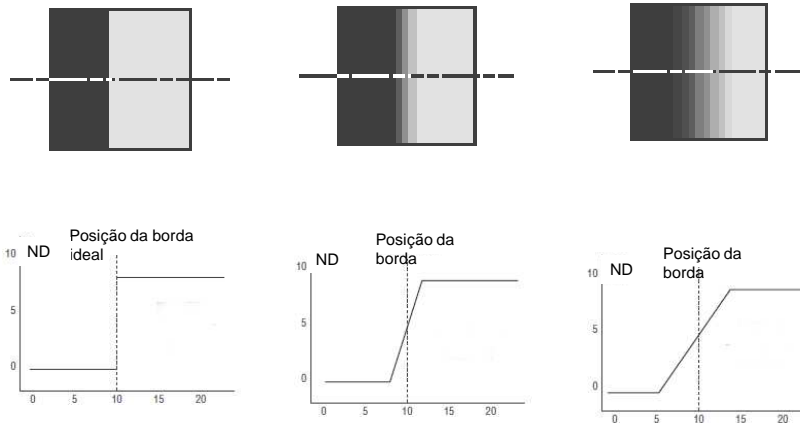


- Detecção de bordas: localizar os pixels das bordas;
- Realce de bordas : aumentar o contraste entre as bordas e o fundo.

Uma imagem em nível de cinza pode ser vista como composta por objetos 3D , em que a 3ª dimensão é o nível de cinza



Bordas e contornos apresentam variações locais de intensidade. Nem sempre as bordas ocorrem na imagem de modo ideal.



Considerando uma função digital (discreta) unidimensional:

- Função f (uma linha de uma imagem digital):

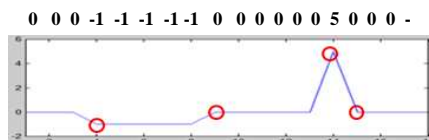


-Derivada de primeira ordem:

-> diferença entre dois NDs vizinhos:

$$f'(x) \approx f(x+1) - f(x).$$

- zero em áreas de intensidade constante;
- diferente de zero ao longo de uma rampa, e no início e fim de uma rampa ou degrau.

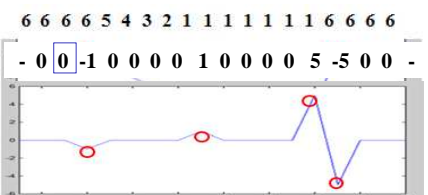


-Derivada de segunda ordem:

-> diferença entre o ND na posição x e seus dois vizinhos:



$$f''(x) \approx f(x+1) - 2f(x) + f(x-1)$$

- zero em áreas de intensidade constante;
- diferente de zero no início e fim de um degrau ou rampa de intensidade;



O gradiente de uma função de duas variáveis $f(x,y)$, nas coordenadas (x,y) é definido por:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

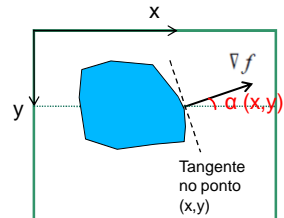
 **bordas horizontais**
 **bordas verticais**

a magnitude deste vetor é:

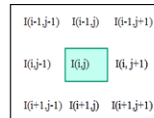
$$|\nabla f| = [G_x^2 + G_y^2]^{1/2}$$

A direcção do vetor gradiente também é uma característica importante para muitas aplicações, ela é dada por:

$$\alpha(x,y) = \arctan \frac{G_y}{G_x}$$



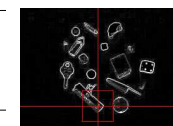
Considerando uma máscara percorrendo a imagem, sendo $I(i,j)$ a posição atual sobre a imagem:



Operador de Robert (1965) usando valores absolutos:

$$GR(i,j) = \text{abs}[I(i,j) - I(i+1,j+1)] + \text{abs}[I(i,j+1) - I(i+1,j)]$$

$$Gr_P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$



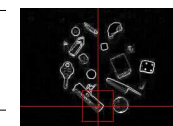
Operador de Prewitt (1966):

$$Gx_P(i,j) = [I(i-1,j+1) + I(i,j+1) + I(i+1,j+1)] - [I(i-1,j-1) + I(i,j-1) + I(i+1,j-1)] ;$$

$$Gy_P(i,j) = [I(i+1,j-1) + I(i+1,j) + I(i+1,j+1)] - [I(i-1,j-1) + I(i-1,j) + I(i-1,j+1)]$$

$$Gx_P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Gy_P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



Operador de Sobel (1970):

$$Gx_S(i,j) = [I(i-1,j+1) + 2I(i,j+1) + I(i+1,j+1)] - [I(i-1,j-1) + 2I(i,j-1) + I(i+1,j-1)] ;$$

$$Gy_S(i,j) = [I(i+1,j-1) + 2I(i+1,j) + I(i+1,j+1)] - [I(i-1,j-1) + 2I(i-1,j) + I(i-1,j+1)]$$

$$Gx_S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Gy_S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



- Bordas **verticais** podem ser detectadas pela **diferença horizontal** entre pontos adjacentes,
- Bordas **horizontais** podem ser detectadas pela **diferença vertical** entre pontos adjacentes.

Algoritmo para determinação de pontos de borda:

Para uma imagem I com dimensão m linhas e n colunas e um limiar T ,

- Percorrer cada posição (i,j) da imagem,
- Em cada posição (i,j) , calcular a magnitude do gradiente $|\nabla f|$, e
- Verificar: se $|\nabla f| > T$, sendo T um limiar, então (i,j) é um ponto da borda.

% Operador de Prewitt

$I = \text{double}(I)$; % imagem com m linhas e n colunas

$G_i = \text{zeros}(m,n)$; % derivada segundo linhas

$G_j = \text{zeros}(m,n)$; % derivada segundo colunas

$\text{Grad} = \text{zeros}(m,n)$; % imagem de saída

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

for $i = 2:m-1$

for $j = 2:n-1$

$G_i(i,j) = [I(i+1,j-1) + I(i+1,j) + I(i+1,j+1)] - [I(i-1,j-1) + I(i-1,j) + I(i-1,j+1)]$;

$G_j(i,j) = [I(i-1,j+1) + I(i,j+1) + I(i+1,j+1)] - [I(i-1,j-1) + I(i,j-1) + I(i+1,j-1)]$;

$\text{Gradiente} = \text{round}(\sqrt{G_i(i,j)^2 + G_j(i,j)^2})$;

if $\text{Gradiente} > 6$ % valor arbitrado

$\text{Grad}(i,j) = 255$;

end

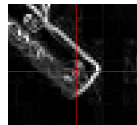
end

end

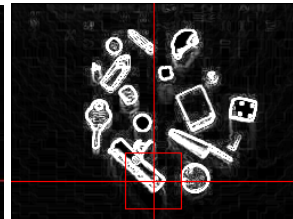
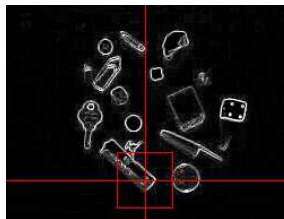
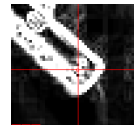
Imagem original



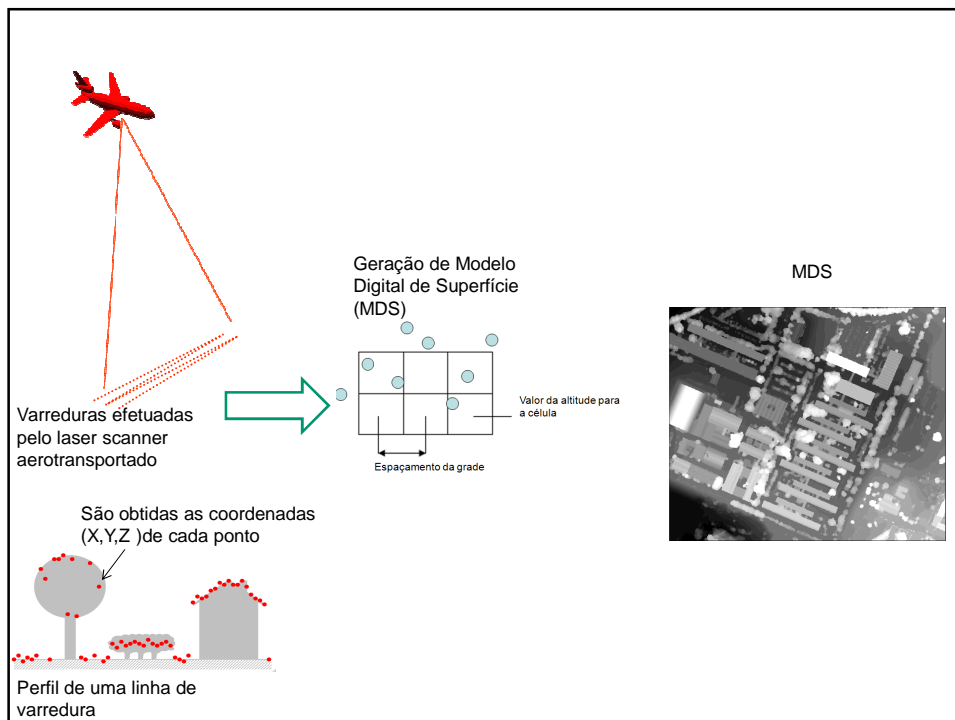
Aplicação do operador de Robert

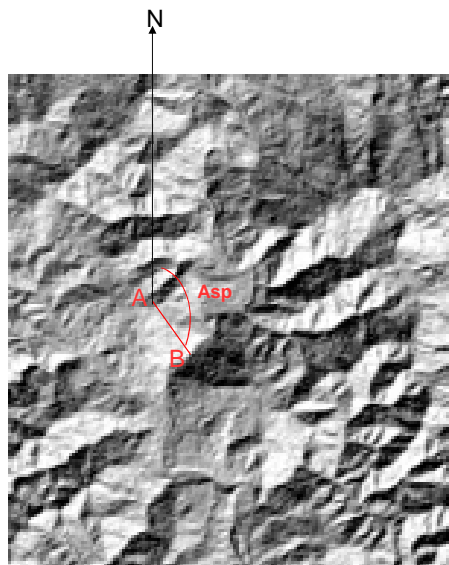


Aplicação do operador de Sobel



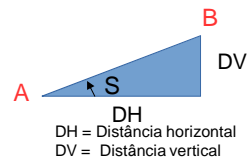
Exercício 3. Implemente o gradiente de Prewitt e determine bordas na imagem *MDS8bits.tif*.





Declividade (S):

$$S = \text{atan}(DV/DH)$$



Aspecto (Asp):

Orientação (Azimute) da face
(começando com o 0 para direção
norte, em graus horário).

Considerando

-*resol* a resolução do modelo de terreno, e
-a vizinhança 8 em relação à posição (i,j):

I(i-1,j-1)	I(i-1,j)	I(i-1,j+1)
I(i,j-1)	I(i,j)	I(i,j+1)
I(i+1,j-1)	I(i+1,j)	I(i+1,j+1)

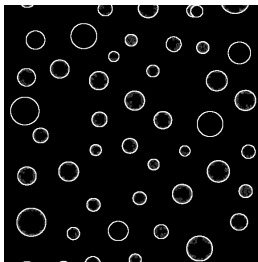
$$b = (I(i-1,j+1) + 2 * I(i,j+1) + I(i+1,j+1) - I(i-1,j-1) - 2 * I(i,j-1) - I(i+1,j-1)) / (8 * \text{resol})$$

$$c = (I(i-1,j-1) + 2 * I(i-1,j) + I(i-1,j+1) - I(i+1,j-1) - 2 * I(i+1,j) - I(i+1,j+1)) / (8 * \text{resol})$$

$$\text{declividade} = \text{atand}(\text{sqrt}(b^2 + c^2))$$

$$\text{aspecto} = \text{atand}(b / c)$$

Declividade

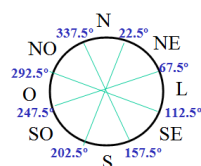


0° sup. plana
90° sup. vertical

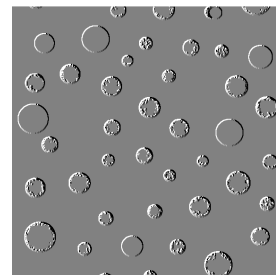
Com as seguintes condições:

Se $c > 0$, aspecto = aspecto + 180°

Se $c < 0$ e $b > 0$, aspecto = aspecto + 360°



Aspecto: Azimute (0° a 360°)



0° preto;
180° cinza médio;
360° branco.

```

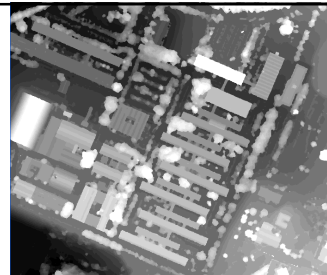
I = double(I);
[m,n] = size(I);
% Gerar imagens de saída:
Dec = zeros(m,n); % Declividade
Asp = zeros(m,n); % Aspecto

for i = 2:m-1
for j = 2:n-1
b = ( I(i-1,j+1) + 2*I(i,j+1) + I(i+1,j+1) - I(i-1,j-1) - 2*I(i,j-1) - I(i+1,j-1) ) / (8*resol);
c = ( I(i-1,j-1) + 2*I(i-1,j) + I(i-1,j+1) - I(i+1,j-1) - 2*I(i+1,j) - I(i+1,j+1) ) / (8*resol);
Dec(i,j) = round(atan(sqrt( b^2 + c^2 )));
if c ~= 0
    Asp(i,j) = round(atan(b /c)) ;
    if c > 0
        Asp(i,j) = 180 + Asp(i,j);
    elseif c < 0 & b > 0
        Asp(i,j) = 360 + Asp(i,j);
    end
end
end
end
end

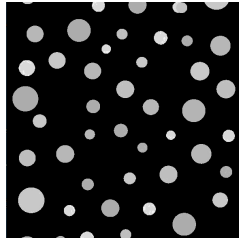
```

Exercício 4:

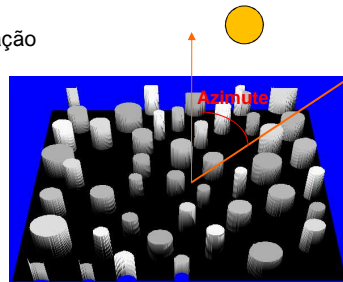
- a) A partir da imagem “MDS8bits.tif “ gerar :
- Declividade
 - Aspecto



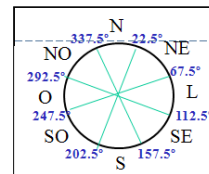
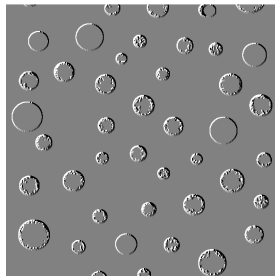
Imagem



Iluminação



Aspecto: Azimute (0° a 360°)



0° preto;
180° cinza médio;
360° branco.

Exercício 5:

A partir da imagem "aspecto " do exercício anterior, saliente a face norte fazendo:

Relação Azimute – valor:

Entre 292,5 e 337,5 -> 100;

Entre 337,7 e 22,5 -> 200;

Entre 22,5 e 67,5 -> 100.

